Naval Ocean Systems Center
San Diego, CA 92152-5000

(4)

Technical Document 1332
September 1988

Temporal Pattern Recognition

C. E. Priebe (NOSC)
C. H. Sung (UCSD)

NAVAL OCEAN SYSTEMS CENTER

San Diego, California 92152-5000

E. G. SCHWEIZER, CAPT, USN
Commander

R. M. HILLYER
Technical Director

ADMINISTRATIVE INFORMATION

Information in this report originally was prepared by members of the Architecture and Applied Research Branch (NOSC Code 421) to be presented as a professional paper and published in the proceedings for IEEE International Conference on Neural Networks held in San Diego, California, 24 to 27 July 1988.

Released by
V. J. Monteleon, Head
Architecture and Applied
Research Branch

Under authority of
J. A. Salzmann, Jr., Head
Information Systems
Division

LO

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) NOSC TD 1332	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Naval Ocean Systems Center 6b. OFFICE SYMBOL (if applicable) NOSC	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State and ZIP Code) San Diego, CA 92152-5000	7b. ADDRESS (City, State and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Naval Ocean Systems Center 8b. OFFICE SYMBOL (if applicable) NOSC	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State and ZIP Code) San Diego, CA 92152-5000	10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	AGENCY ACCESSION NO.
62232N	CDB2	RC32S11	DN306 242

11. TITLE (include Security Classification)

TEMPORAL PATTERN RECOGNITION

12. PERSONAL AUTHOR(S)

C. E. Priebe (NOSC) and C. H. Sung (UCSD)

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Presentation/speech	FROM July 1988 TO July 1988	September 1988	12

16. SUPPLEMENTARY NOTATION

17. COSATI CODES		
FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)

command/control systems,
data fusion,
electro-optical,
opto-electronic devices.

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

A self-organizing network architecture for the learning of recognition codes corresponding to temporal patterns is described. The problem of temporal pattern recognition has been studied via both conventional pattern recognition methods and neural network approaches for more than twenty years. The problem presents itself in many real-world situations. In any non-trivial environment in which a proposed system will function the spectre of temporal information-information coming into the system over a period of time-is evident. In many cases it is not sufficient to process the information independent of its relative time-order. Disciplines as diverse as speech recognition, robotics and data fusion/situation analysis require that the temporal aspect of the data be considered. In temporal environments such as these the information lost when using a non-temporal approach can be prohibitive. This approach is formulated to make use of this important temporal information.

The network described herein takes as its input individual incoming events. Sequences of these events (letters, phonemes, or, more abstractly, object sightings in a vision system), received by the system over time are categorized as specific sequences by the temporal system. The temporal system produces Gaussian classifications that represent the statistics of the temporal data, and the system uses a noisy environment, giving as output a Gaussian distance from the stored sequence, thus providing an analog measure of "closeness of fit" to currently known patterns. The system can recognize sequences with missing or extraneous elements, as well as out-of-order sequences. In addition, a desirable prediction property – the system realizes it may be in a particular sequence long before the entire sequence has been introduced – is a consequence of the multi-dimensional gaussian distance calculation.

Presented at IEEE International Conference on Neural Networks, 24-27 July 1988, San Diego, CA.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a. NAME OF RESPONSIBLE PERSON C. E. Priebe	22b. TELEPHONE (include Area Code) (619) 553-4048 22c. OFFICE SYMBOL Code 421

DD FORM 1473, 84 JAN

83 APR EDITION MAY BE USED UNTIL EXHAUSTED
ALL OTHER EDITIONS ARE OBSOLETE

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

Temporal Pattern Recognition

Carey E. Priebe **
Dr. Chen-Han Sung *

*Center for Artificial Intelligence and Neural Networks
San Diego State University
San Diego, CA 92182
and
Department of Computer Science and Engineering
University of California at San Diego
La Jolla, CA 92093

**Architecture and Applied Research Branch
Code 421
Naval Ocean Systems Center
San Diego, CA 92152

ABSTRACT

A self-organizing network architecture for the learning of recognition codes corresponding to temporal patterns is described. The problem of temporal pattern recognition has been studied via both conventional pattern recognition methods and neural network approaches for more than twenty years. The problem presents itself in many real-world situations. In any non-trivial environment in which a proposed system will function the spectre of temporal information - information coming into the system over a period of time - is evident. In many cases it is not sufficient to process the information independent of its relative time-order. Disciplines as diverse as speech recognition, robotics and data fusion / situation analysis require that the temporal aspect of the data be considered. In temporal environments such as these the information lost when using a non-temporal approach can be prohibitive. This approach is formulated to make use of this important temporal information.

The network described herein takes as its input individual incoming events. Sequences of these events (letters, phonemes or, more abstractly, object sightings in a vision system), received by the system over time are categorized as specific sequences by the temporal system. The temporal system produces Gaussian classifications that represent the statistics of the temporal data, and the system uses a learning scheme of moving mean and moving covariance to update these self-developed classes. The system recognizes sequences in a noisy environment, giving as output a Gaussian distance from the stored sequence, thus providing an analog measure of "closeness of fit" to currently known patterns. The system can recognize sequences with missing or extraneous elements, as well as out-of-order sequences. In addition, a desirable prediction property - the system realizes it may be in a particular sequence long before the entire sequence has been introduced - is a consequence of the multi-dimensional Gaussian distance calculation.

I. Temporal Patterns

The ability to understand one's environment, an essential property in the elusive search for intelligence, is not governed solely by static pattern recognition. The order in which events occur can be even more important than the events themselves, and an intelligent system, whether it be a mouse or a robot, must be able to detect and understand this ordering. Thus the dimension of time allows access to a wealth of information about the current environment, past events, and expectations about the future. An ability to incorporate time into information processing is necessary for abilities such as the recognition of sequences of events, understanding cause and effect, making predictions and planning.

The ability to recognize sequences is essential for many tasks, most notably those involved with audition and vision. A sequence may consist of a stream of phonemes, typed letters or frames from a movie. Once the initial preprocessing has been done and the individual members of the sequence have been recognized, the task is shifted. The processing is then concerned with determining which of the known sequences are represented by the input. Since the answer may depend on the context in which the input has been received, the system must return all the sequences that the input might represent with a confidence rating indicating the quality of match between the input and the known sequences.

Consider a stream of phonemes. The task is to recognize the words that are being spoken. Here it is important to recognize the order in which the phonemes appear and the words these sequences of phonemes might represent. A certain amount of error will appear in the sample due to normal fluctuations in a speaker's voice and a large number of variable conditions in the environment, and this must be dealt with. Also, a subsequence may be a legitimate word which the system should recognize in order to allow a more sophisticated system to deal with the ambiguities.

A simple formulation of this recognition problem, similar to that provided by Tank & Hopfield[4], is shown in Figure 1. The problem is to extract known sequences from noisy data. Ideally, we must be able to recognize the sequence "I D A H O" from the stream given, despite the fact that a perfect match is not present. The letters in the figure can be thought of as abstract events and the words as higher-level activities, or sequences of events.

To process this stream of events the system must be able to represent order information and work on imperfect exemplars. The duration of the constituents of the sequence and the spacing between them is also important. The system must be able to incorporate this information, and ideally it should learn the sequences and adapt to the environment in which it is operating.

II. Temporal Information Processing System (TIPS)

A. Overview

The Temporal Information Processing System (TIPS) proposed is a multi-layer network architecture which is distinctly different from conventional neural network paradigms. The network self-organizes during the learning phase, developing Gaussian categories for the sequences being learned. These Gaussian categories, represented as individual nodes in the F4 level, are based on the activation level of the F3 field. As input stimuli enter the system, the F3 field experiences a decay factor, providing for the ordering of its nodal activations based on their time of input [3]. The temporal system, a Gaussian classifier, processes the static input stimuli using a combination of this temporal decay and moving mean and covariance [2] to obtain a representation for the statistics of the input patterns and update the categorizations. The system attempts to classify the F3 representation of stimuli received thus far into an existing temporal pattern category. Failing this, the system creates a new category for the current sequence. These categories can then, in parallel, perform independent, local distance calculations when presented a novel input, thereby determining proper categorization(s) for new input stimuli.

B. Gaussian Theory

The F4 field utilizes a Gaussian classification scheme to achieve an unsupervised partitioning of the input space. Each node consists of a multi-dimensional Gaussian activation function in which the mean and covariance matrix adapt to the data. The system then learns the statistics of the data by representing the data as a sum of multi-dimensional normal distributions.

In one extreme, in which each input represents a distinct class and each node learns just one data point, this subsystem produces a Voronoi classifier [1]. The Voronoi classifier for a set of points is the optimal nearest neighbor classifier for the points. In this case, the system is nothing more than a nearest neighbor classifier, returning the Gaussian distance from each of the stored

points.

In the other extreme, in which all the points are classified by a single node, the system fits a normal distribution to the data. In this case the system would compute the mean and covariance matrix for the inputs and its output would be a normal distribution with those parameters.

A large class of distributions can be approximated by a mixture of Gaussians. Therefore the temporal system approximates the distribution of its input by a collection of Gaussians at the F4 level. This allows for a distributed system in that it uses many nodes to represent the distribution of the input. Since the support of the Gaussians is infinite, there is a degree of redundancy in this representation. The system degrades gracefully under nodal failures, yielding the fail-safe property that is desirable in many applications.

The individual Gaussians are represented by the nodes in the F4 field. Each Gaussian may be of a different dimensionality. The covariance for a particular Gaussian is represented by the activation function of the node, while the mean can be represented in the connections from F3 to F4 (see Figure 2). The Gaussians are presented with the n-dimensional input from field F3 having n nodes and, in parallel, compute their respective activation values (Gaussian distances) from this input. The activation value for these Gaussian nodes in F4 can be obtained via the following formula:

$$a_j(\underline{x}) = \frac{\exp\left(-0.5[(\underline{x} - \underline{\mu})^t \cdot \Sigma^{-1} \cdot (\underline{x} - \underline{\mu})] \right)}{(2\pi)^{(d/2)} * |\Sigma|^{(1/2)}} .$$

Here $a_j(\underline{x})$ is the activation value of the j^{th} Gaussian node G_j when presented with vector input $\underline{x}$. Σ is the covariance matrix for Gaussian node G_j, while $\underline{\mu}$ is the vector-valued mean for this Gaussian. d is the dimensionality of this particular Gaussian and is equated with the number of non-zero input connections to node G_j. Since the components of the mean $\underline{\mu}$ are represented as a node's input weights, these weights can be thought of as shifting the origin for the node. A Gaussian function is then applied to the inputs of the node, as opposed to a sigmoid (see Figure 3).

C. Moving Mean and Covariance

Only those nodes whose activation values reach some threshold defined by that node's activation function (usually the value of the Gaussian at one standard deviation from the mean) are considered to be a likely category for the current input and are updated. This updating consists of moving the mean and variance of the Gaussian based on the current input and some measure of the total number of inputs to the Gaussian thus far. For the updating of the mean, we have

$$\mu(j+1) = \mu(j) + [1/(j+1)][I(j+1) - \mu(j)]$$

where $\mu(j)$ is the (one-dimensional) mean after the j^{th} input and $I(j+1)$ is the $j+1^{st}$ input to be categorized in this Gaussian. The updating of the covariance is similar. Here

$$S_{xy}(j+1) = S_{xy}(j) + [j/(j+1)] * A_{xy}(j+1) ,$$

with

$$A_{xy}(j+1) = [(x(j+1) - \mu_x(j)) * (y(j+1) - \mu_y(j))] \quad .$$

Then

$$\Sigma_{xy}(j+1) = \frac{1}{j} S_{xy}(j+1) \quad ,$$

where we are calculating $\Sigma_{xy}(j+1)$, the x,y component of the covariance matrix Σ after the $j+1^{st}$ input clustered in this Gaussian. $x(j)$ and $y(j)$ are the x and y component of the input vector after the j^{th} input, and μ_x, μ_y are components of the mean.

Independent of this learning procedure, the values across the F4 field represent the relative likelihoods that the current input belongs to a particular class. In the unsupervised case, this is the solution, and the system, having been presented with a given sequence, will be able to recognize that sequence, along with similar sequences that have missing or additional features. In fact, the activation value of an F4 node is a measure of how close a given input is to previously learned sequences. If a particular Gaussian is allowed to process on only those inputs that are a part of its make-up, i.e. Gaussians only process within their dimensionality, a fundamental way is established to ensure that extra or spurious input, as found in a noisy environment, does not affect the recognition of learned sequences.

D. System Dynamics

The workings of the TIPS system can be described in three sections: the decay factor at the F3 layer, the connections between F3 and F4, and the activation function at F4.

1. Decay at F3

When input stimuli are received into the F3 field, the activation values in the field experience a decay factor that acts as an ordering function. By using this decay (as seen in Figure 4) the F3 field can develop a representation in which the order the stimuli were received is preserved. This ordering can then be used to determine the distance of the current F3 field from the learned patterns already represented on the Gaussian level.

It is clear that the F3 field must contain more than a single node for each distinct stimuli. For example, if the system is categorizing sequences of letters, there must be more than one F3 node corresponding to the letter "A". Were this not the case, the second "A" entering the system would activate the only F3 node corresponding to "A", overwriting information concerning the previous "A".

2. F3 - F4 Connections

The connections between F3 and F4 represent the components of the mean for the Gaussians at F4 (as in Figure 2). During learning the Gaussian nodes incorporate patterns from the F3 level into their incoming connections. This is done by setting the connection weight equal to the activation value of the pre-synaptic F3 node when a new F4 node is being allocated. When an existing Gaussian is being updated, the moving mean calculation described in Section II (C) is used. This weight is then the offset, or mean, to be used in the distance computation when calculating the F4 activations.

3. Activation at F4 -- Distance Calculation

The actual implementation of the F4 field deviates slightly from the pure Gaussian scheme. A number of the alterations are designed to allow the system to run in real-time while maintaining the ability to recognize patterns under varying environmental conditions (as in Section III). The basic distance calculation is altered, but a multi-dimensional distance calculation is still at the heart of the scheme. The equation for activation in the F4 layer at time t + 1 is given below.

$$G_j(t+1) = \Delta G_j(t) + B_j * D_j * e^{-\eta E_j}$$

This equation gives the activation of of Gaussian node G_j at time $t+1$ in terms of the node's previous activation, $G_j(t)$, and an exponential function of E_j, the error between the input stimulus and the mean of Gaussian G_j. Δ is a short-term memory constant which allows an added dimension of history - the node's previous activation - to be incorporated into the activation calculation. η is a Gaussian parameter that alters the default standard deviation of the Gaussian function. B_j is a statistical parameter based on the number of patterns seen. This parameter can give an a priori estimate of the probability that the current stimulus belongs to the category represented by node G_j. In its simplest form, B_j equates to the ratio of the total number of patterns seen to the number of patterns that have been categorized as belonging to the category indicated by node G_j.

D_j is a function of the dimension of the current stimulus and the (static) dimension of Gaussian G_j. Since not all Gaussian nodes will have the same dimensionality, and because we require the system to begin to recognize a pattern prior to receiving the entire pattern, and hence the entire dimensionality of the pattern, it is necessary to include a weighting factor into the activation function of the Gaussian nodes. Because this dimensionality parameter is a function of the current input stimulus' dimensionality, the system is able to determine that it does not have a perfect match at node G_j even in the case where there is no error in the intersection of the input dimensions with the dimension of node G_j. A side effect of using this dimensionality parameter is that a Gaussian node need not be aware of stimuli outside its dimensionality. Hence each node is inherently oblivious to noise in the environment, and focuses attention only on its intended pattern.

This activation calculation at the F4 level gives the system its output - the higher the activation value for node G_j, the more certain the system is that it is seeing the pattern associated with node G_j. The fact that each node performs its activation calculation independently of the other Gaussian nodes yields an inherently parallel network structure.

In addition, the system gives as its output the activation value, or belief value, for each pattern independently. If the input stimuli indicate two seperate patterns are being seen, the system will yield two seperate Gaussian nodes with relatively high activation values, independent of one another. This property is akin to the ability to carry all possible hypotheses while waiting for complete input. The system need not choose a subset of all possibilities to pursue based on incomplete data, as many rule-based systems are forced to do, but rather has all possible Gaussian nodes independently attempting to validate the existance of their individual patterns.

III. System Capabilities and Results: Temporal Pattern Recognition

Evaluation of the performance of any temporal pattern recognition system is far from a straightforward task. In many, if not most, instances the incoming data does not fit exactly with any of the learned sequences. There is no value in a yes/no decision on the presence of a sequence. The system is asked instead to give a "best guess" of which pattern(s) it is seeing based on some set of criteria. This is by definition an ambiguous task, and the ultimate result can only be

evaluated by looking at the criteria and the input data and attempting to generate a "better" solution, as defined by the human evaluator.

Nevertheless, this section attempts to document the proper performance of the TIPS system in a variety of differing environments. Figure 1 depicts the system recognizing input stimuli as known patterns with varying degrees of certainty. In the figure, the y-axis represents the activation values for the Gaussian nodes corresponding to individual patterns. Both the relative and the absolute magnitudes of these nodal activations are significant in evaluating the system. The x-axis represents the input stimuli, received over time, upon which the system is processing. These stimuli are represented in the figures as letters, but are more properly thought of as abstract events.

The simplest and most straightforward experiment that can be used to test the quality of the TIPS system is that of sequence recognition in a noiseless environment. Here it can be determined whether the system has performed properly or improperly. As a complete sequence is input it exactly matches one of the learned patterns, and it is imperative that the system correctly identify this pattern. In addition, the system must exhibit the ability to indicate the presence of multiple patterns. Multiple patterns are ultimately flagged as present due to the relative magnitude of their activations compared to the activations of the other patterns stored in the system. This is vital since many application environments do not ensure mutually exclusive events or patterns. The system gives as output more than one pattern with a high value. This represents the fact that the Gaussian nodes corresponding to each of the indicated patterns received input close to their corresponding means, and therefor the activation values for these nodes are close to the maximum possible value the Gaussian can attain.

In evaluating the system under different stimuli conditions, it is useful to understand and assess the prediction ability of the system. Prediction is the capability to indicate the presence of a pattern prior to receiving the entire pattern. A system with little or no prediction capability is of limited use, since the system user would like to be warned of possible happenings prior to their conclusion, thus allowing the user to affect the ultimate outcome by acting, rather than reacting, to stimuli. The prediction capabilities are relevant in both noisy and noiseless environments. Figure 1 shows TIPS indicating the possible presence of a pattern prior to receiving the entire pattern. "idaho" is flagged as potentially present prior to the input of the final letter(s) in the pattern, as is "ohio".

Here the Gaussian nodes are indicating that the input stimulus is close to the mean in the intersection of the dimensionalities of the input stimulus and the stored pattern. This fact may indicate that the specified pattern is beginning to show itself, and the system must be able to recognize this fact. However, the system should not indicate that it is certain of the presence of such a pattern, regardless of a perfect match in this intersection of dimensionalities, if the input stimulus lacks a significant fraction of the stored pattern's dimensionality.

A temporal pattern recognition system must be able to recognize patterns in an environment in which data is missing. Figure 1 illustrates the ability of the TIPS system to indicate a particular pattern despite the fact that the input stimuli consists of only a partial pattern. "utah" is indicated with some small degree of certainty despite the absence of two characters that the system has been taught belong in the pattern.

Missing data may be caused by sensor inadequacies or by subtle variations in the actual pattern itself which alter the sequence in some small way, yet leave the overall meaning of the pattern unaltered. Although the pattern received does not correlate exactly with the learned pattern, TIPS must indicate that the input stimuli is similar to one of the learned patterns. There is no hard and fast rule for determining how certain the system should be that it is seeing a given pattern. The only definitive statement that can be made is that the system must give some indication that it is close to seeing the learned pattern. TIPS actually outputs a "distance" from the learned Gaussian mean to the received stimuli, using this as its analog output.

In the case of extraneous data, or noise, TIPS is asked to ignore the noise where possible and process only the relevent information to determine the existance of learned patterns. The individual Gaussians are concerned only with inputs that are represented in their dimensionality, and therefore ignore stimuli that are not a part of the Gaussian domain. For stimuli present in the Gaussian dimensionality, the Gaussian nodes attempt to determine which input of a particular class

(or dimension) is closest to the component of the mean for that dimension. In this way, extraneous data is disregarded.

Figure 1 illustrates TIPS processing patterns with extraneous data. TIPS recognizes the presence of the pattern "idaho" despite having spurious stimuli ("w" and "i") included in the input. It is important to understand that not only does the system have to deal with the spurious stimuli, but there is also a time-warping implicitly introduced by this extraneous input. For the "idaho" pattern, the system has been taught to expect the "d" immediately following the "i". When "d" follows at a significantly longer interval, the pattern itself is warped, and the system must be able to recognize a pattern regardless of this phenomenon. This time-warping is quite evident in the recognition of "ohio" in Figure 1. There is a significant delay between the initial "o" and the rest of the pattern ("hio"). This delay, more than the extraneous stimuli ("iwda") degrades the system's beleif in the presence of "ohio". Nevertheless, we desire the system to indicate the possibility of the pattern being present.

TIPS can be altered to perform at various stages along a continuum from order being all-important to disregarding order. Figure 1 illustrates TIPS processing out-of-order stimuli in a situation where order is deemed "somewhat important". The pattern "iowa" is indicated by the input sequence "oiwda". The recognition of "iowa" is degraded slightly due to the spurious "d", but the out-of-order stimuli "o" and "i" cause the system to be less certain of the pattern's existance. Nevertheless, the system indicates the possibility of the pattern being present.

The extent to which the system can identify patterns despite the stimuli being received out-of-order rests on a design decision tightly tied to the type of environment in which the system operates. Since there exists no generic criteria for determining how important ordering should be, the system needs to be flexible in this regard.

IV. Summary

The need for more complete temporal knowledge processing has been clear to researchers in artificial intelligence and neural network theory for more than twenty years. Work in rule based systems has failed to yield a satisfactory approach. In addition, much of the neural network research in the area has been devoted to a simple transformation of temporal data into a spatial pattern. Although this is the approach which lends itself to early small-scale success, it is necessary to use the incoming temporal data in a way that preserves the knowledge inherent in this data. The different aspects of temporal information -- short-term, medium-term and long-term context -- indicate a separate approach to extracting the knowledge for each aspect. The system proposed herein performs processing on incoming data without first depriving it of much of the information of importance. The system learns the statistics of its input and adapts to a changing environment. Although specific architectures may vary, the component concepts described above allow a system to utilize the available temporal information to more fully understand its environment.

V. References

1. Aggarwal, Chazelle & Guibas, Parallel Computational Geometry (1985), *Proceedings of the 26th IEEE Symposium on the Foundations of Computer Science*, 468-477.
2. Chan, Golub, & LeVeque, Algorithms for computing the sample variance: analysis and recommendations (1983), *The American Statistician*, (37) 242-247.
3. Grossberg (1986) in Pattern Recognition by Humans and Machines, Vol 1, eds. Schwab & Nusbaum, Academic Press, 216-226.
4. Tank & Hopfield, Neural computation by concentrating information in time (1987), *Proc. Natl. Acad. Sci.* USA 84, 1896-1900.